基于加权 TextRank 的中文自动文本摘要 *

黄 波，刘传才

(南京理工大学计算机科学与工程学院，南京 210094)

摘 要：现有的中文自动文本摘要方法主要是利用文本自身的信息，其缺陷是不能充分利用词语之间的语义相关等信息。鉴于此，提出了一种改进的中文文本摘要方法。此方法将外部语料库的信息用词向量的形式融入到 TextRank 算法，通过 TextRank 与 word2vec 的结合，把句子中每个词语映射到高维词库形成句向量。充分考虑句子之间的相似度、关键词的覆盖率和句子与标题的相似度等因素，以此计算句子之间的影响权重，并选取排序最靠前的句子重新排序作为文本的摘要。在本文的数据集中取得了较好的效果。实验结果表明，此方法自动提取中文摘要的效果比原方法好。

关键词：文本摘要；TextRank；词向量；句子相似度

中图分类号：TP391.1 doi: 10.19734/j.issn.1001-3695.2018.07.0528

Chinese automatic text summarization based on weighted TextRank

Huang Bo, Liu Chuancai

(*School of Computer Science & Engineering, Nanjing University of Science & Technology, Nanjing* 210094, *China*)

Abstract: The method of Chinese existing automatic text summarization mainly utilized the text's own information, and its defect was that it cannot make full use of the related semantic information between the words. Therefore, this paper proposed an improved Chinese text summarization method. This method integrated the information of the external corpora into the TextRank algorithm in the form of a word vector. Combined TextRank with Word2vec, it mapped each word in the sentence to the high-dimensional lexicon to form a sentence vector. This method fully considered the similarity between sentences, the coverage of keywords and the similarity between sentence and title to calculate the influence weights among sentences, and choose the top-ranked sentences used as the summarization of the text. This method has achieved good results in the data set of this paper. The results of experiment show that this method is more effective than the original method in extracting Chinese summarization automatically.

Key words: text summarization; TextRank; word vector; sentence similarity

0 引言

随着计算机信息技术的发展，互联网上的各种信息数据以指数级速度爆炸增长。如何从海量的文本信息中快速获得用户所需要的信息变得格外重要，传统人工提取信息的方法已经不能满足需求，而自动文本摘要越来越受到关注，具有很大的应用价值。

文本摘要通过对文本信息概括总结提取出主要内容。根据摘要方式的不同，自动文摘技术可以分为抽取式摘要和生成式摘要两种[1]。1958 年，IBM 公司的 Luhn 基于高频词语的评分提出了一种文本摘要方法[2]，开启了自动文本摘要研究的先河。葛斌等人通过把文本中的句子构建成无向图，将文本摘要的提取转换为图模型中节点的权重计算[3]。Erkan 等人[4]提出了一种基于 LexRank 算法的文本摘要算法，主要根据词的权重或者句子的特征计算句子的权重，利用向量空间模型表示成图模型，通过计算句子之间的相似度提取出相似度较大的句子作为文本摘要。李峰等人[5]使用关键字扩展的方法从新闻文本中自动提取摘要。本文的研究内容主要是面向单文档的中文文本，即针对单独的一篇文档基于文本中句子的权重评分提取句子生成摘要。

1 相关工作

文本摘要可以利用文本信息本身的内容和结构特征实现，并在一定程度上满足需求，以 TextRank 算法[6]为典型代表。除此之外，也可以通过大量的语料信息进行训练学习来抽取摘要。这类方法不同于传统算法实现简单，需要大量的训练数据。对于一篇文档，传统算法大多忽略它的词语语义、语法等要素，简单地当成是词语的集合，并且每个词语都是独立出现的，互相不依赖彼此之间出现与否。如果将外部知识如语料库等信息融入到自动文本摘要的算法之中，理论上能够改善效果。由 Google 研究团队开发的 word2vec 模型[7]使用词向量[8]表示词语，可用来表示词语之间的关系。本文将 word2vec 与 TextRank 算法进行融合并加以改进，采用基于词向量的高维词库映射计算句子之间的相似度，而取代基于相同词语共同出现的频率作为句子之间的影响权重，但弱化了共现词语的加权作用。Luhn 的论文[2]指出，频繁出现的词语与文章的主题有比较大的关联，根据词语出现的频率计算句子的权重并排序形成摘要，准确率比不少复杂的方法要高[9]。李峰等人[5]基于 TextRank 使用关键词扩展提取文本摘要取得了优于原方法的效果。关键词对文章中的摘要句子的提取起着很大的作用，增加关键词的覆盖率，即关键词在句

收稿日期：2018-07-22；**修回日期：**2018-09-14 **基金项目：**国家自然科学基金资助项目（61373062，61373063，61473155）

作者简介：黄波（1993-），男，安徽安庆人，硕士，主要研究方向为数据挖掘（chnhuangbo@qq.com）；刘传才（1963-），男，教授，博导，主要研究方向为模式识别、计算机视觉.

子分词后所有词语中占的比例作为句子的加权权重，补充了词语的加权作用。除此之外文章标题往往在一定程度上代表了文章的主要内容，程园等人[10]充分考虑文本中的词频、标题、句子位置及句子相似度等特征构建特征加权函数提取关键句生成摘要。句子与文本标题相似程度越大越可能是关键句。因此本文利用句子之间的相似度、句子中关键词的覆盖率和句子与标题的相似度共同作为句子之间权重的影响因子，以提取文本的摘要结果。

1.1　TextRank 算法

经典的 TextRank 算法将 Google 公司 PageRank[11, 12]算法的思想引入到了文本摘要之中，基于 TextRank 算法的自动文本摘要算法将文本信息拆分成句子作为网络节点，并组成句子网络的图模型，用来表示句子之间的结构关系。并通过图的迭代计算实现重要性排序。该方法不需要对语料库或者其他相关文档提前进行学习训练，实现简单且效果不错，因此得到了广泛的应用。

TextRank 算法的一般模型可以表示为一个带权的图模型 $G=(V, E)$，其中 V 为节点集合，即句子构成的节点集合，E 为边集合，用 w_{ji} 表示任意两个节点 V_i 和 V_j 之间边的权重，即句子 V_i 与句子 V_j 之间的相似度。对于任意给定的节点 V_i，$In(V_i)$为指向该节点的节点集合，$Out(V_i)$为节点 V_i 指向的节点集合[13]。节点 V_i 的评分数值计算公式如下：

$$WS(V_i)=(1-d)+d*\sum_{V_j\in In(V_i)}\frac{w_{ji}}{\sum_{V_k\in Out(V_j)}w_{jk}}WS(V_j) \tag{1}$$

式(1)为 TextRank 的递归式[14]，其中 d 为阻尼系数（Damping Factor），取值范围为 0 到 1 之间，表示图模型中某节点指向其他节点的概率。阻尼系数过大会使需要迭代的次数骤增且算法的排序不稳定，阻尼系数过小会导致迭代过程没有明显效果，一般情况下取值为 0.85[15]。

边的权重 w_{ji} 用句子的相似度来表示，基于计算句子之间共同词语的覆盖率，即通过比较不同句子之间共同词语出现的个数。对于给定两个句子 S_i 和 S_j，采用如下公式进行计算：

$$w_{i,j}=Similarity(s_i,s_j)=\frac{|\{t_k\,|\,t_k\in S_i\wedge t_k\in S_j\}|}{\log(|S_i|)+\log(|S_j|)} \tag{2}$$

其中 $S_i=[w_{i,1}, w_{i,2}, \ldots, w_{i,j}, \ldots, w_{i,n}]$，为句子去除停用词之后的词语集合，$w_{i,j}$ 为第 i 句中去除停用词后第 j 个词语。PageRank 算法通过计算两个网页之间的互相引用次数得到网页的重要程度，在 TextRank 算法中则是用句子的相似度来取代网页之间相互链接的个数[13]。比如“我/爱/中国”和“你/喜欢/中国/吗”，边的权值为：

$$w_{i,j}=Similarity(s_i,s_j)=\frac{1}{\log(3)+\log(4)}。$$

这种方法在计算句子的相似度起到了一定程度上的效果，但是却忽略了词语的语法、语义等影响因素，如近义词、反义词之间的关系等。

使用 TextRank 算法计算图模型中各节点的得分时，首先指定图模型中每个节点任意的初始值，然后根据边的权重递归迭代计算，直到图模型中任意节点的误差率小于预先设定的极限值时收敛，每个节点的最后得分不受给定初始值的影响。前人的研究实验表明，一般取极限值为 0.0001[5]时，递归计算能够很好地收敛。

1.2　Word2vec 模型

Word2vec[7]是 2013 年 Google 公司开发的用于词向量计算的一款工具，它可以高效地训练百万以上级别的数据集，主要有以 Huffman 树为基础的 CBOW（continuous bags-of-words model）和 skip-gram（continuous skip-gram model）两种模型。CBOW 模型基于上下文预测当前词语的概率，而 Skip-gram 模型则是基于当前词语预测上下文的概率，两种模型都包含 input、projection 和 output 三层结构[16]，分别如图 1、2 所示。

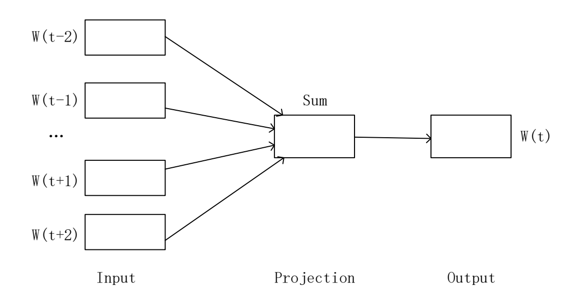


图 1　crow 模型示意

Fig. 1　CROW model

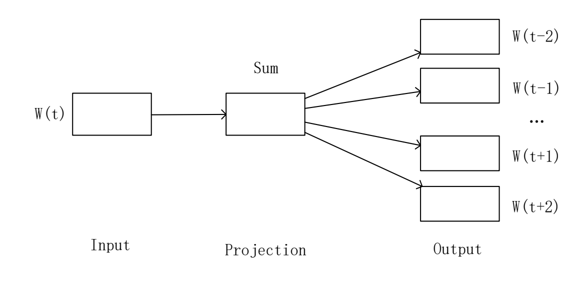


图 2　Skip-gram 模型示意

Fig. 2　Skip-gram model

两种方法利用人工神经网络训练大批量文本，将文本中的词语转换为 N 维向量空间中的词向量，利用计算空间文本向量的相似度衡量文本的相似度。当神经网络训练完成时，可求出语料库中词的出现次数超过预先设定值的词向量。

2　研究方法

基于 TextRank 的自动文本摘要算法的思想是将文本摘要的提取过程转换成文本中句子重要程度的排序过程。首先根据 word2vec 模型训练语料库得到词语的词向量转换获得句向量，然后根据句向量计算句子之间的相似度，构建候选句子网络的图模型，即完整的句子之间的概率转移矩阵，通过迭代运算获取节点的重要性，实现自动文摘的排序和抽取。

2.1　中文文本预处理及特征选择

利用自然语言处理（natural language processing，NLP）[17]相关技术对文本信息进行处理，首先将文本正文切分成单个句子，然后再利用中科院 NLPIR 汉语分词系统（又名 ICTCLAS2013）将句子进行分词[18]，过滤掉文本中无意义的停用词，如“的、地、得、了”等，得到每个句子的词集合。

2.2　基于词向量模型的中文文本表示

Word2vec 本质上是利用浅层的神经网络模型（一般为三层）学习训练词语在语料库或数据集中出现的概率，将词语用一个合适的维度空间表示成数值形式，即词向量，词语之间的相似性可以使用词向量之间的相似性来度量。相较基于传统稀疏矩阵表示词语的方法在解决实际问题时经常会遇到维数灾难，无法表示词语之间的语义、语法信息和内在联系等问题，word2vec 生成的词向量不仅解决了维度灾难问题——即词向量的维度会因为文本的增加无限制地增大，而且

在从大规模语料库中挖掘了词语之间的关联属性，因此提高了词向量语义上的准确度。Word2vec 在大规模语料库中训练学习得到的词向量中蕴涵的词语语义信息，可以很好地用来表示词语之间蕴涵的联系。比如，传统模型中“好像”和“似乎”两个词语毫无联系，但在 Word2vec 词向量中两个词语有着较高的相似度。

一般常使用维度为 100 作为训练 Word2vec 词向量的维度标准，如果维度过大，模型的训练复杂度将会剧增。并且词向量每一个维度上的数值只能表示该维度上词语正相关或者负相关的程度，其数值大小并不能表示与训练词库中词语对应的实际相关程度。所以在默认维度为一百的数量级情况下，使用词向量直接累加求平均值或者取每个维度的最大值这两种方法都不能很好地用来表示句子。本文利用训练的词向量模型，设计了一种句向量的计算方法，进行句子相似度的计算。日常汉语的使用中，几千至几万个词语几乎能够表达出绝大多数文本的信息。本文采用词库映射的方法，构建高维词库将文本的语义信息映射到常用的高频词库中，利用词库映射句子语义得到句向量。

首先构建一个具有 N 个词的高频常用词词库，结合词库中每个词的词向量，可以将每个文本映射成一个具有 N 维的向量，向量的每一维分别是高频词库中该维度所对应的词语和文本中每个词语的相似度的最大值，文本可以是句子或者文章等等。在词语的相似度计算上，这种方法相比基于上下文的方法，每一个词语都得到了该词语在高维词表中的映射，并表达成了 Word2vec 模型的稠密特征，形如[0.231262, 0.178923, 0.798699, 0.325891,…]，几乎不存在维度为 0 的稀疏情况。可以赋予每个词语更加丰富的语义分布，本质上是对 bag of words 的一个扩展，而不是形如[0,0,1,0,0,0,0,0,…]用简单的非 0 或 0 来表示绝对相关或绝对不相关，这对于解决 bag of words 的稀疏性问题效果较好。

使用 word2vec 工具利用高维词库 R 表示文本向量，假设高维词库中共有 n 个词语，表示词向量的形式：

$R=[r_1, r_2, \ldots, r_i, \ldots, r_n]$，$r_i$ 表示高维词库中第 i 个词语的词向量。

设文本经过分词去掉特殊符号和停用词后，有 m 个词语，则使用词向量将文本表示为：

$T=[t_1, t_2, \ldots, t_j, \ldots, t_m]$，$t_j$ 表示分词后文本中第 j 个词语的词向量。

将文本映射到高维词库中则表示为：

$$\begin{aligned} S=[&\max_{1\le j\le m}\left(similarity\left(r_1,t_j\right)\right),\\ &\max_{1\le j\le m}\left(similarity\left(r_2,t_j\right)\right),\ldots,\\ &\max_{1\le j\le m}\left(similarity\left(r_i,t_j\right)\right),\ldots,\\ &\max_{1\le j\le m}\left(similarity\left(r_n,t_j\right)\right)] \end{aligned} \tag{3}$$

其中：采用余弦距离表示向量之间的相似度，即

$$similarity\left(r_i,t_j\right)=cos\left(r_i,t_j\right) \tag{4}$$

具体算法如下：

算法 1 求中文句子文本的句向量

输入：句子文本，高维词库词向量集合 R，词向量模型 model

输出：句向量

```
对句子文本分词、去除符号和停用词，得到词语集合 D
初始化句子词向量集合 T
for d in D
   if(d in model)//如果词语 d 包含在模型 model 中
   t = model[d]//获得词语 d 对应的词向量
   else
   t = [0]
    end if
    将 t 加入到词向量集合 T
end for
初始化句子向量集合 S
for r in R
  初始化词语相似度集合 C
  for t in T
        将 cos(r,t)加入 C
     end for
   取集合 C 中最大的值加入集合 S
end for
return S
```

其中，第 15 步的 cos(r,t)即用式(4)求词向量之间的相似度，第 19 步中需要将集合类型进行转换成数值类型。

2.3 TextRank 权值计算

原始的 TextRank 自动文本摘要算法计算句子之间的相似度一般采用式(2)，但是只是简单地计算句子之间相同词语的覆盖率作为边的权重。为了进一步提高文本摘要的效果，本文使用 Word2vec 工具将外部知识引入到自动文摘中，在计算 TextRank 算法中边的权重上作出如下改进。利用文档句向量之间的关系，在句子之间构建 TextRank 模型，通过句子之间的相似度、关键词的覆盖率和句子与标题的相似度对 TextRank 图模型节点之间的概率进行加权，计算每个句子的影响力权重，按照权重由大到小排序。

2.3.1 句子之间的相似度

将句子使用中文分词去掉标点符号和停用词之后得到词语的集合，按照式(3)将句子映射到高维词库中表示成向量形式。计算两个使用高维词库映射的句子 S_i 和 S_j 相似度时，也采用余弦距离表示（S_i 和 S_j 都是使用高维词库映射的向量）：

$$W_s\left(S_i,S_j\right)=similarity\left(S_i,S_j\right)=\cos\left(S_i,S_j\right) \tag{5}$$

2.3.2 关键词的覆盖率

句子中包含的关键词越多，则句子的重要程度越高。$W_k(S_i, S_j)$表示句子节点 S_i 关键词覆盖率权重传递给句子节点 S_j 的权重，公式为

$$W_k\left(S_i,S_j\right)=\frac{P\left(S_j\right)}{\sum_{S_k\in Out(S_i)}P\left(S_k\right)} \tag{6}$$

式(6)中，$p(S_i) = len(keywords(S_i))/len(S_i)$，表示句子 S_i 中关键词的个数与句子中总词数（去除标点符号和停用词）的比例，即句子 S_i 中关键词的覆盖率。

2.3.3 句子与标题的相似度

句子与文本标题的相似度越高，则句子的重要程度越高。$W_t(S_i, S_j)$表示句子 S_i 节点与标题相似度权重传递给句子节点 S_j 的权重，公式为

$$W_t\left(S_i,S_j\right)=\frac{\text{similarity}(S_j,S_t)}{\sum_{S_k\in Out(S_i)}\text{similarity}(S_k,S_t)} \tag{7}$$

S_t 为映射到高维词库的表示文本标题的词向量，S_i 和 S_j 为映射到高维词库的句子词向量，$Similarity(S_j, S_t)$即为句子 S_j 与标题 S_t 的相似度。计算其相似度一般采用余弦距离，即式(5)。

根据上述公式构建新的句子之间影响力的权重，将 $W_s(S_i,S_j)$、$W_k(S_i,S_j)$、$W_t(S_i,S_j)$

三种权重影响因子分别归一化之后得到

$$W_s^{'}(S_i,S_j)、W_k^{'}(S_i,S_j)、W_t^{'}(S_i,S_j)$$

构建句子影响力权重公式为

$$w_{ij}=aW_s^{'}\left(S_i,S_j\right)+bW_k^{'}\left(S_i,S_j\right)+cW_t^{'}\left(S_i,S_j\right) \quad (8)$$

其中：a、b、c 分别代表句子之间的相似度、关键词的覆盖率和句子与标题的相似度计算权重时所占的比重，即对于构建好的图模型，提出影响节点（句子）之间权重的三个影响因子，a、b、c 为这三种句子之间权重影响因子在归一化后的加权系数，加权系数越大代表其对应的权重影响因子在计算权重时的影响力越大。其取值均在 0~1，并且 $a+b+c=1$。

3　实验

3.1　实验数据及评价标准

本文采用 2017 年 10 月发布的维基百科中文数据和清华大学自然语言处理实验室推出的中文文本新闻数据集中的一部分，过滤掉标点符号和其他无关符号等数据清洗之后，通过中科院张华平博士研究的中文分词工具 NLPIR 汉语分词系统（又名 ICTCLAS）进行分词，形成提供学习训练的文本数据集，然后使用基于 python 语言的自然语言处理库 Gensim 中 Word2vec 模块，采用 CBOW 模型、维度为 100、窗口大小为 5 等默认参数对该文本数据集进行学习训练得到词向量模型文件[13]。

目前中文自动文本摘要没有一个公认的评估语料和评估标准，本文的测试文本数据集来自于清华大学自然语言处理实验室推出的中文文本新闻数据集，取其中的体育、财经、科技、时政、娱乐的 5 个类别各 20 篇共 100 篇作为测试语料库。由三位语言学相关专业的研究生对测试语料中的文本信息人工提取摘要，三位研究生分别独立地从每篇文档中提取出 8 到 10 个摘要句子，并按与文章内容相关程度从大到小排序，最后综合三人的摘要结果，取结果相同的与文章内容相关程度最大的 3 个句子作为人工摘要句子。

实验摘要质量的评价方法采用自动摘要领域使用最广泛的 Rouge 指标[19]，Rouge 基于摘要中 n 元词（*n*-gram）的共现信息来评价摘要，是一种面向 n 元词召回率（recall）的自动化评价方法。基本思想是将系统自动生成的自动摘要与人工生成的标准摘要对比，通过统计两者之间重叠的基本单元（n 元语法、词序列和词对）的数目来评价摘要的质量。本文采用 Rouge-1、Rouge-2、Rouge-L 三种评价指标来评价。

3.2　影响因子加权系数的确定

为了合理评估自动文摘的质量，本文采取上述的 Rouge-1、Rouge-2、Rouge-L 三种评价指标作为衡量标准，计算每篇文本每个句子中基于句子之间的相似度、关键词的覆盖率和句子与标题的相似度三个权重影响因子的加权系数。综合句子之间权重影响因子的加权系数 a、b、c（其中 $a+b+c=1$），本文取 a、b、c 以 0.05 的间距改变（增大或减小，保证 $a+b+c=1$），经过大量实验，计算不同加权系数组合下的 Rouge-1、Rouge-2、Rouge-L 值，选取了一部分有代表的实验数据如表 1 所示。

针对上文选取的 10 组参数组合，分别计算每篇测试文本的自动文摘结果，并取其均值，实验结果如图 3 所示。

实验结果表明，当 $a=0.6$、$b=0.2$、$c=0.2$ 时，自动文摘的效果最好，当 a 的值逐渐接近 0.6 时，评价效果总体呈上升趋势，而当 a<0.5 时，评价效果逐渐下降，可以看出，句子之间的相似度在 TextRank 权值计算中起到了重要的作用，关键词的覆盖率和句子与标题的相似度相对重要程度较小。

表 1　不同比例组合下的加权系数值

Table 1　Weighting coefficient values under different ratio combinations

组数	*a*	*b*	*c*
1	1	0	0
2	0.9	0.05	0.05
3	0.8	0.1	0.1
4	0.7	0.2	0.1
5	0.6	0.3	0.1
6	0.6	0.2	0.2
7	0.6	0. 25	0.15
8	0.55	0.35	0.1
9	0.5	0.3	0.2
10	0.4	0.3	0.3

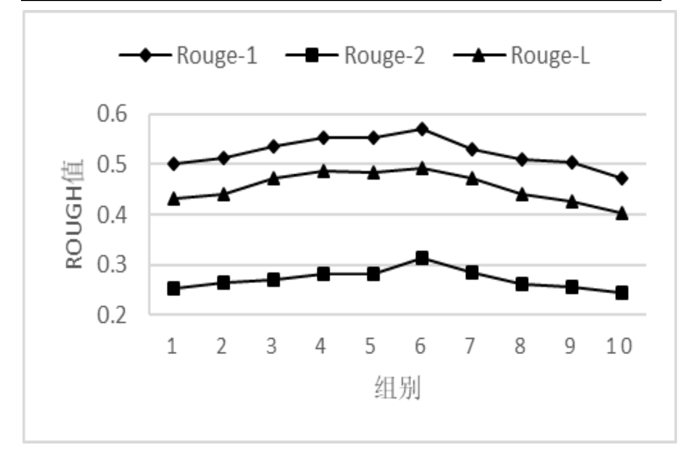


图 3　不同参数组合的实验结果

Fig. 3　Experimental results of different parameter combinations

3.3　实验结果及分析

通过实验本文得到了一组最佳的加权系数组合，为了验证本文方法的有效性，分别采用基于词频-逆文档概率（TF-IDF）[20]的方法、基于 LexRank、基于 TextRank 和本文改进的方法对 100 篇测试文本摘要数据集进行实验对比。实验结果如图 4 所示。

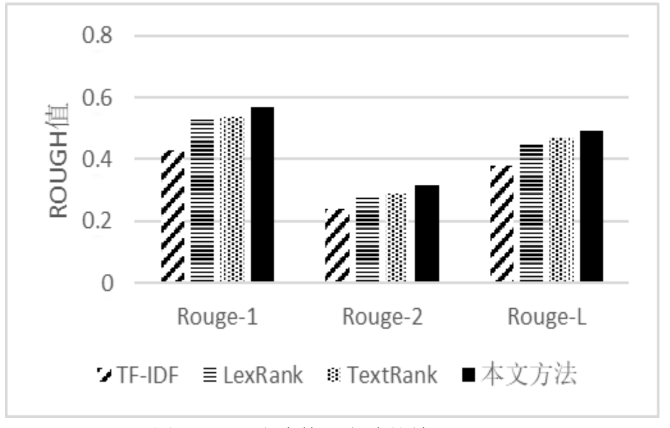


图 4　不同文本摘要方法的效果对比

Fig. 4　Comparison of the effects of different text summarization methods

以上实验数据表明，本文所改进的方法在 Rouge-1、Rouge-2 和 Rouge-L 3 个评价指标上均有了明显的提高。基于 TF-IDF 的方法相比而言效果最差，本文改进的方法要优于 LexRank 算法和传统的 TextRank 文本摘要算法，除了考虑文本自身特征外，还引入外部知识库，增加了句子权重的影响因素。

4 结束语

本文提出了一种基于词向量加权的中文自动文本摘要算法，基于图模型的句排序算法结合 word2vec 模型的词向量，充分考虑了句子之间的相似度、关键词的覆盖率、句子与标题的相似度等因素的影响。实验结果表明，针对单文档的中文自动文摘，与传统的 TextRank 算法比较，本文方法文摘的抽取效果更好，有效提高了自动文摘的质量。但文本摘要速度有待提升，这是下一步改进的目标。

参考文献：

[1] Gambhir M, Gupta V. Recent automatic text summarization techniques: a survey [J]. Artificial Intelligence Review, 2016, 47(1): 1-66.

[2] Luhn H P.The automatic creation of literature abstracts [J]. IBM Journal of Research & Development, 1958, 2 (2): 159-165.

[3] 葛斌，李芳芳，李阜，等. 基于无向图构建策略的主题句抽取 [J]. 计算机科学, 2011, 38(5): 181-185. (Ge Bin, Li Fangfang, Li Fu, *et al*. Subject Sentence Extraction Based on Undirected Graph Construction [J]. Computer Science, 2011, 38(5): 181-185.)

[4] Erkan, G, Radev D R.. LexRank: graph-based lexical centrality as salience in text summarization [J]. Journal of artificial intelligence research, 2004, 22 (1): 457-479.

[5] 李峰，黄金柱，李舟军，等. 使用关键词扩展的新闻文本自动摘要方法 [J]. 计算机科学与探索, 2016, 10 (3): 372-380. (Li Feng, Huang Jinzhu, Li Zhoujun, *et al*. Automatic summarization method of news texts using keywords expansion [J]. Journal of Frontiers of Computer Science and Technology, 2016, 10 (3): 372-380.)

[6] Yu ShanShan, Su Jindian, Li Pengfei, *et al*. Towards high performance text mining: a textrank-based method for automatic text summarization [J]. International Journal of Grid & High Performance Computing, 2016, 8 (2): 58-75.

[7] Goldberg Y, Levy O. word2vec Explained: deriving Mikolov *et al*. 's negative-sampling word-embedding method [EB/OL]. (2014) [2018-09-13]. https://arxiv. org/abs/1402. 3722v1.

[8] Yang Zhitong, Zheng Jun. Research on Chinese text classification based on Word2vec [C]//Proc of the 2nd IEEE International Conference on Computer and Communications.2017: 1166-1170.

[9] Kumar Y J, Goh O S, Halizah Basiron, *et al*. A review on automatic text summarization approaches [J]. Journal of Computer Science, 2016, 12 (4): 178-190.

[10] 程园，吾守尔·斯拉木，买买提依明·哈斯木. 基于综合的句子特征的文本自动摘要 [J]. 计算机科学, 2015, 42(4): 226-229. (Chen Yuan, Wushouer Silamu, Maimaitiyiming Hasimua, *et al.* Automatic test summarization based on comprehensive characteristics of sentense [J]. Computer Science, 2015, 42(4): 226-229.)

[11] Haveliwala T H. Topic-sensitive PageRank: a context-sensitive ranking algorithm for Web search [J]. IEEE Trans on Knowledge & Data Engineering, 2003, 15 (4): 784-796.

[12] Chen Ningyuan, Nelly Litvak, Mariana Olvera - Cravioto. Generalized PageRank on directed configuration networks [J]. Random Structures & Algorithms, 2017, 51 (2): 237-274.

[13] 夏天. 词语位置加权 TextRank 的关键词抽取研究 [J]. 现代图书情报技术, 2013, 29(9): 30-34. (Xia Tian. Study on Keyword Extraction Using Word Position Weighted TextRank [J]. New Technology of Library and Information Service, 2013, 29(9): 30-34.)

[14] Mihalcea R, Tarau P. TextRank: bringing order into texts [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2004: 404-411.

[15] Brin S, Page L. The anatomy of a large-scale hypertextual web search engine [J]. Computer networks and ISDN systems, 1998, 30 (1-7): 107-117.

[16] Lilleberg J, Zhu Yun, Zhang Yanqing. Support vector machines and Word2vec for text classification with semantic features [C]//Proc of the IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing. Beijing: IEEE, 2015: 136-140.

[17] Sun Shiliang, Luo Chen, Chen Junyu. A review of natural language processing techniques for opinion mining systems [J]. Information Fusion, 2017, 36: 10-25.

[18] Chen Xinchi, Qiu Xipeng, Zhu Chenxi, *et al.* Long Short-term memory neural networks for chinese word segmentation [C]//Proc of Conference on Empirical Methods in Natural Language Processing. 2015: 1197-1206.

[19] Lin C Y, Eduard H. Automatic evaluation of summaries using N-gram co-occurrence statistics [C]//Proc of Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Stroudsburg: Association for Computational Linguistics, 2003: 71-78.

[20] Paik Jiaul H. A novel TF-IDF weighting scheme for effective ranking [C]//Proc of International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2013: 343-352.